

Package: lightson (via r-universe)

September 21, 2026

Title Download and Analyse Nighttime Lights Satellite Data

Version 0.1.1

Description Downloads and processes nighttime lights (NTL) satellite data from NASA VIIRS/Black Marble (daily, monthly, and annual composites) and India's ISRO Bhuvan NTL portal.

License MIT + file LICENSE

Depends R (>= 4.1.0)

Encoding UTF-8

Roxygen list(markdown = TRUE)

Imports htr2 (>= 1.0.0), terra (>= 1.7.0), sf (>= 1.0.0), jsonlite, digest

Suggests geodata, ggplot2, patchwork, scales, httptest2, pkgdown, withr, testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

VignetteBuilder knitr

URL <https://lightson.saketlab.org>,
<https://github.com/saketlab/lightson>

BugReports <https://github.com/saketlab/lightson/issues>

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository <https://saketlab.r-universe.dev>

Date/Publication 2026-09-13 18:04:33 UTC

RemoteUrl <https://github.com/saketlab/lightson>

RemoteRef HEAD

RemoteSha 321901e9a4ad607052cf66662d3c1e50f94f1b20

Contents

bhuvan_raster	2
bhuvan_stats	3
cache_clear	4
earthdata_token	5
extract_panel	6
get_india_admin	7
ntl_download	8
ntl_index	9
ntl_source_agreement	10
plot_ntl_map	10
plot_ntl_panel	12
plot_ntl_trend	13
theme_ntl	14
Index	15

bhuvan_raster	<i>Download Bhuvan NTL raster tiles for a region</i>
---------------	--

Description

Fetches annual nighttime lights GeoTIFF tiles from the Bhuvan WMS service for a given spatial extent and converts the RGB visualization output to a single-band luminance raster.

Usage

```
bhuvan_raster(  
  region,  
  years,  
  force = FALSE,  
  width = 1024,  
  height = 1024,  
  bbox_pad = c(0, 0, 0, 0)  
)
```

Arguments

region	An sf object, SpatVector, or ISO 3166-1 alpha-3 country code (e.g., "IND"). Used to derive the WMS bounding box.
years	Integer vector of years to download (2012-2024).
force	Logical. If TRUE, re-download even if cached. Default FALSE.
width, height	Pixel dimensions of the WMS request. Default 1024x1024.
bbox_pad	Numeric vector of length 1 or 4 (xmin, ymin, xmax, ymax sides) in degrees, added to the region bounding box before the WMS request. Useful when GADM boundaries clip disputed territories (e.g. bbox_pad = c(0, 0, 0, 3.5) extends the north edge by 3.5 degrees to include all of Jammu and Kashmir).

Details

The Bhuvan WMS returns RGB visualization rasters. Luminance is computed as $lum = 0.2126 \cdot R + 0.7152 \cdot G + 0.0722 \cdot B$ (ITU-R BT.709). Values are comparable across years for trend analysis but differ in units from VIIRS nW/cm²/sr. Do not mix Bhuvan luminance and VIIRS radiance in the same regression without normalising first.

Value

A named list of single-band SpatRaster objects, keyed by year (e.g., `list("2020" = rast(...), "2021" = rast(...))`). Errors if the WMS fetch fails for any requested year.

Collection break

Bhuvan NTL data from 2024 onwards is derived from VNP46A4 **Collection 2.0**, while 2012-2023 uses the legacy **Collection 1.0**. The two collections use different calibration and atmospheric correction algorithms, so luminance values are not directly comparable across this boundary.

Examples

```
## Not run:
districts <- sf::read_sf("https://bharatviz.org/India-bhuvan-districts.geojson")
rasters <- bhuvan_raster(districts, years = 2018:2023)
panel <- extract_panel(rasters, districts)

## End(Not run)
```

bhuvan_stats	<i>Get pre-aggregated Bhuvan NTL state statistics</i>
--------------	---

Description

Wraps the ISRO Bhuvan NTL REST API to return mean radiance per Indian state across years.

Usage

```
bhuvan_stats(years = NULL, state = NULL)
```

Arguments

years	Integer vector of years to return. NULL (default) returns all available years (currently 2012-2024).
state	Character. State name as returned by the API (e.g., "BIHAR"). NULL (default) returns all states.

Details

For district-level or custom-geometry statistics, use `bhuvan_raster()` together with `extract_panel()`.

Value

A tidy data frame with columns `state` (character), `year` (integer), and `mean_radiance` (numeric). State names are normalised to title case (e.g., "Tamil Nadu" not "TAMILNADU").

Collection break

Bhuvan NTL data from 2024 onwards is derived from VNP46A4 **Collection 2.0**, while 2012-2023 uses the legacy **Collection 1.0**. The two collections use different calibration and atmospheric correction algorithms, so radiance estimates are not directly comparable across this boundary. Requests spanning both collections trigger a warning.

Examples

```
## Not run:
# All states, all years
df <- bhuvan_stats()

# Bihar only
df <- bhuvan_stats(state = "BIHAR")

# Multiple years
df <- bhuvan_stats(years = 2015:2023)

## End(Not run)
```

cache_clear	Clear the local lightson cache
-------------	--------------------------------

Description

Removes cached raster files downloaded by `ntl_download()` or `bhuvan_raster()`.

Usage

```
cache_clear(source = NULL)
```

Arguments

`source` One of "viirs", "bhuvan", or NULL.

Value

Invisibly returns the number of files removed.

earthdata_token	<i>Get NASA Earthdata bearer token</i>
-----------------	--

Description

Returns the bearer token used for VIIRS/Black Marble downloads from NASA LAADS DAAC.

Usage

```
earthdata_token(token = NULL)
```

Arguments

token	Character string. If NULL, checks EARTHDATA_TOKEN env var, then tries EARTHDATA_USERNAME + EARTHDATA_PASSWORD.
-------	--

Details

Getting a token (web):

1. Create an account at <https://urs.earthdata.nasa.gov>
2. Accept the MERIS and Sentinel EULAs under Profile > EULAs
3. Go to Profile > **Generate Token**, click "Generate Token", then "Show Token"
4. Copy the token (starts with EDL-) and store it

Supplying the token:

- Set EARTHDATA_TOKEN environment variable (recommended: add to ~/.Renviron)
- Or pass it directly as the token argument
- Or set EARTHDATA_USERNAME + EARTHDATA_PASSWORD to fetch it automatically

Bhuvan NTL functions ([bhuvan_stats\(\)](#), [bhuvan_raster\(\)](#)) do not require any authentication.

Value

The access token string, invisibly.

Examples

```
## Not run:
# Option 1: token from Profile > Generate Token on urs.earthdata.nasa.gov
Sys.setenv(EARTHDATA_TOKEN = "EDL-xxxxxxxxxxxxxxxxxxxxx")
token <- earthdata_token()

# Option 2: username + password (fetches token automatically)
Sys.setenv(EARTHDATA_USERNAME = "myuser", EARTHDATA_PASSWORD = "mypass")
token <- earthdata_token()
```

```
# Option 3: pass directly
token <- earthdata_token("EDL-xxxxxxxxxxxxxxxxxxxx")

## End(Not run)
```

extract_panel

Compute zonal statistics for a panel of rasters

Description

Takes a named list of `SpatRaster` objects (one per time period, as returned by `ntl_download()` or `bhuvan_raster()`) and an `sf` or `SpatVector` polygon layer, and returns a tidy long-format data frame with one row per region and time period. Annual names such as "2025" produce the original four-column output. ISO-date names used by monthly and daily Black Marble additionally produce a date column.

Usage

```
extract_panel(rasters, polygons, fun = "mean", id_col = NULL)
```

Arguments

<code>rasters</code>	A named list of <code>SpatRaster</code> objects. Names should be years or ISO dates (e.g., "2020" or "2025-11-12").
<code>polygons</code>	An <code>sf</code> object or <code>SpatVector</code> containing the regions to summarise over.
<code>fun</code>	Aggregation function name passed to <code>terra::zonal()</code> . Default "mean". Also accepts "sum", "median", etc.
<code>id_col</code>	Name of the column in <code>polygons</code> to use as <code>region_id</code> . If <code>NULL</code> (default), the first character column is used.

Details

The `polygons` argument accepts any `sf` object: GADM, `bharatviz` GeoJSONs, LGD districts, or any shapefile loaded with `sf::read_sf()`.

Value

A data frame with columns:

region_id Character. Region identifier from `polygons`.

year Integer. Year, taken from the name of each raster.

mean_radiance Numeric. Aggregated radiance value.

n_pixels Integer. Number of non-NA cells contributing to the stat.

Examples

```
## Not run:
districts <- sf::read_sf("https://bharatviz.org/India-bhuvan-districts.geojson")
rasters <- bhuvan_raster(districts, years = 2018:2023)
panel <- extract_panel(rasters, districts)
head(panel)

## End(Not run)
```

get_india_admin	<i>Get Indian administrative boundaries</i>
-----------------	---

Description

Returns Indian administrative boundaries as an sf object. Useful as the polygons argument to [extract_panel\(\)](#).

Usage

```
get_india_admin(
  level = c("country", "state", "district"),
  source = c("gadm", "bhuvan")
)
```

Arguments

level	One of "country", "state", or "district".
source	One of "gadm" (default) or "bhuvan". For level = "state", the default uses a bundled LGD dataset with SoI-compliant boundaries regardless of source. "bhuvan" fetches geometries from the Bhuvan API so that names match bhuvan_stats() output exactly.

Details

State boundaries use a bundled LGD (Local Government Directory) dataset derived from the Survey of India.

For custom boundaries, load your own file with `sf::read_sf()` and pass it directly to [extract_panel\(\)](#).

Value

An sf object.

Examples

```
## Not run:
states <- get_india_admin("state")
districts <- get_india_admin("district")

# custom boundary:
districts <- sf::read_sf("https://bharatviz.org/India-bhuvan-districts.geojson")

## End(Not run)
```

ntl_download

Download NASA Black Marble nighttime lights

Description

Downloads calibrated VIIRS Black Marble annual (VNP46A4), monthly (VNP46A3), or daily (VNP46A2) rasters from NASA LAADS DAAC.

Usage

```
ntl_download(
  region,
  years = NULL,
  token,
  force = FALSE,
  product = c("annual", "monthly", "daily"),
  dates = NULL,
  quality = c("high", "all")
)
```

Arguments

region	An sf object, SpatVector, or ISO 3166-1 alpha-3 code.
years	Integer years. Required for annual data; selects all twelve months for monthly data when dates is omitted.
token	NASA Earthdata bearer token from earthdata_token() .
force	Re-download cached output.
product	One of "annual", "monthly", or "daily".
dates	Date-like vector. Required for daily data. Monthly dates are normalised to the first day of their month.
quality	For daily data, "high" masks pixels whose mandatory QA flag is nonzero; "all" retains every retrieval.

Value

Named list of cropped SpatRaster objects. Names are years for annual data and ISO dates otherwise.

Examples

```
## Not run:
token <- earthdata_token()
annual <- ntl_download("IND", 2020:2025, token)
monthly <- ntl_download("IND", 2025, token, product = "monthly")
daily <- ntl_download("IND",
  token = token, product = "daily",
  dates = as.Date("2025-11-12")
)

## End(Not run)
```

ntl_index

*Index nighttime-light values to a baseline period***Description**

Converts each group's values to an index where its baseline observation is 100. This is useful for comparing changes between sources with different units, but does not make their absolute levels comparable.

Usage

```
ntl_index(
  panel,
  baseline,
  group_cols = NULL,
  time_col = "year",
  value_col = "mean_radiance",
  index_col = "ntl_index"
)
```

Arguments

panel	Data frame containing nighttime-light observations.
baseline	Baseline value in time_col, such as 2020 or an ISO date.
group_cols	Grouping columns. If omitted, uses source when present plus region_id or state.
time_col	Time column name. Defaults to "year".
value_col	Value column name. Defaults to "mean_radiance".
index_col	Name for the new index column.

Value

panel with an additional numeric index column.

ntl_source_agreement	<i>Measure agreement between two nighttime-light panels</i>
----------------------	---

Description

Aligns observations by region and time, then calculates within-region correlation between two sources. Correlation compares co-movement, not absolute levels or units.

Usage

```
ntl_source_agreement(  
  x,  
  y,  
  id_col = "region_id",  
  time_col = "year",  
  value_col = "mean_radiance",  
  method = c("pearson", "spearman", "kendall")  
)
```

Arguments

- x, y Panel data frames, such as outputs from `extract_panel()`.
- id_col Region identifier column.
- time_col Shared time column.
- value_col Measurement column in both panels.
- method Correlation method passed to `stats::cor()`.

Value

Data frame with region, correlation, and paired observation count.

plot_ntl_map	<i>Plot a nighttime lights raster map</i>
--------------	---

Description

Renders a single SpatRaster as a map. Two palettes are available:

Usage

```
plot_ntl_map(
  raster,
  polygons = NULL,
  title = "Nighttime lights",
  caption = NULL,
  palette = c("glow", "bhuvan"),
  dark = TRUE,
  limits = NULL
)
```

Arguments

raster	A single-layer SpatRaster. Pass one element from the list returned by <code>ntl_download()</code> or <code>bhuvan_raster()</code> .
polygons	Optional sf object for boundary overlay.
title	Plot title. Default "Nighttime lights".
caption	Optional caption string.
palette	One of "glow" (default) or "bhuvan".
dark	Logical. If TRUE, the plot background is #080c14 (deep navy) and text is light. Ignored when palette = "bhuvan" (always dark bg). Default TRUE.
limits	Numeric vector of length 2 for the continuous scale. Only used when palette = "glow". NULL uses the raster min/max. Set explicitly when comparing multiple panels on the same scale.

Details

- "glow" (default): continuous dark-olive-to-cream-white ramp modelled on the NASA Black Marble composite. Suited to dark backgrounds and multi-panel year comparisons.
- "bhuvan": 4-class categorical legend matching the Bhuvan portal: black (< 5 nW/cm2/sr), dark blue-grey (5-25), cream (26-200), red (> 200).

Value

A ggplot2 object.

Examples

```
## Not run:
rasters <- bhuvan_raster("IND", years = 2022, bbox_pad = c(0, 0, 0, 3.5))
plot_ntl_map(rasters[["2022"]], dark = TRUE)
plot_ntl_map(rasters[["2022"]], palette = "bhuvan")

## End(Not run)
```

plot_ntl_panel	<i>Plot a multi-year nighttime lights panel</i>
----------------	---

Description

Arranges one map per year into a patchwork grid with a shared title, subtitle, and caption. A shared luminance scale is computed from the 99.5th percentile across all rasters so panels are directly comparable.

Usage

```
plot_ntl_panel(
  rasters,
  polygons = NULL,
  title = "India at night",
  subtitle = NULL,
  caption = NULL,
  palette = c("glow", "bhuvan"),
  ncol = 3L,
  bg_colour = "#080c14",
  title_colour = "#CDD9E5",
  subtitle_colour = "#8B949E",
  year_colour = "#E8A87C"
)
```

Arguments

rasters	Named list of SpatRaster objects as returned by ntl_download() or bhuvan_raster() , keyed by year.
polygons	Optional sf object for boundary overlay.
title	Overall plot title. Default "India at night".
subtitle	Optional subtitle string.
caption	Optional caption string.
palette	One of "glow" (default) or "bhuvan". Passed to plot_ntl_map() .
ncol	Number of columns in the grid. Default 3.
bg_colour	Background colour of the assembled panel. Default "#080c14" (deep navy).
title_colour	Title text colour. Default "#CDD9E5".
subtitle_colour	Subtitle and caption text colour. Default "#8B949E".
year_colour	Year label colour on each individual map. Default "#E8A87C" (warm amber).

Details

Requires the **patchwork** package.

Value

A patchwork object.

Examples

```
## Not run:
rasters <- bhuvan_raster("IND", years = c(2012, 2015, 2018, 2021, 2024))
states <- get_india_admin("state")
plot_ntl_panel(rasters,
  polygons = states, title = "India at night, 2012-2024",
  caption = "Source: ISRO Bhuvan NTL portal"
)

## End(Not run)
```

plot_ntl_trend	<i>Plot nighttime lights radiance trends</i>
----------------	--

Description

Line chart of mean radiance over time for one or more regions. Accepts output from [extract_panel\(\)](#) or [bhuvan_stats\(\)](#).

Usage

```
plot_ntl_trend(
  panel,
  region = NULL,
  title = "Nighttime lights radiance",
  subtitle = NULL,
  caption = NULL
)
```

Arguments

panel	A data frame as returned by extract_panel() (with <code>region_id</code> column) or bhuvan_stats() (with <code>state</code> column), plus <code>year</code> and <code>mean_radiance</code> .
region	Character vector of region identifiers to plot. <code>NULL</code> (default) plots all regions.
title	Plot title. Default "Nighttime lights radiance".
subtitle	Optional subtitle string.
caption	Optional caption string.

Value

A `ggplot2` object.

Examples

```
## Not run:
panel <- extract_panel(rasters, districts)
plot_ntl_trend(panel, region = "Lucknow")
plot_ntl_trend(panel, region = c("Mumbai", "Delhi", "Chennai"))

## End(Not run)
```

theme_ntl

ggplot2 theme for nighttime lights plots

Description

A transparent-background `theme_minimal()` base tuned for NTL data: tight grid, bold title, and optional overrides via `...`. Ink colour is a single equal-contrast grey (#767676, ~4.5:1 against both pure black and pure white) so the same rendered figure reads on light and dark pages alike, since pkgdown bakes each figure to one static PNG at build time.

Usage

```
theme_ntl(base_size = 12, ...)
```

Arguments

<code>base_size</code>	Base font size. Default 12.
<code>...</code>	Additional <code>ggplot2::theme()</code> arguments.

Value

A `ggplot2` theme object.

Index

bhuvan_raster, [2](#)
bhuvan_raster(), [3](#), [5](#), [6](#), [11](#), [12](#)
bhuvan_stats, [3](#)
bhuvan_stats(), [5](#), [7](#), [13](#)

cache_clear, [4](#)

earthdata_token, [5](#)
earthdata_token(), [8](#)
extract_panel, [6](#)
extract_panel(), [3](#), [7](#), [10](#), [13](#)

get_india_admin, [7](#)
ggplot2::theme(), [14](#)

ntl_download, [8](#)
ntl_download(), [6](#), [11](#), [12](#)
ntl_index, [9](#)
ntl_source_agreement, [10](#)

plot_ntl_map, [10](#)
plot_ntl_map(), [12](#)
plot_ntl_panel, [12](#)
plot_ntl_trend, [13](#)

stats::cor(), [10](#)

terra::zonal(), [6](#)
theme_ntl, [14](#)